1. **The problem the paper trying to solve**

The primary problem the paper addresses is the computational inefficiency caused by activation recomputation in large transformer models. During backpropagation in deep networks, activations from the forward pass are required, but storing them for very deep models like transformers can consume a significant amount of memory. Current methods involve recomputing these activations, but this approach increases the computational load. The goal of the paper is to reduce the need for this recomputation, thereby lowering both memory and computational requirements.

1. **What is the impact of the work**

This work is crucial for improving the scalability of transformer models, particularly in improving training efficiency for extremely large models like GPT-3 and BERT variants. By reducing the need for activation recomputation, the approach allows for more efficient use of memory and computational resources. This is particularly impactful for hardware-limited environments and enables training larger models without requiring a significant increase in hardware. The work improves model training efficiency, making it highly relevant for research labs and industry, where the high cost of training large models is a significant bottleneck.

1. **The main proposed ideas**

Sequence parallelism and selective activation recomputations along with the tensor model parallelism are the two main ideas proposed in this paper to reduce the need for activation rec-computation.

1. **Summary of different components**

* Sequence parallelism involves distributing the sequence dimension across multiple GPUs for layers that consumes significant storage but less computation resources like layer normalization, drop-out layer etc
* Selective activation recomputation applies for the layers like attention layers that take up considering amount of memory and are not computationally expensive. The technique leverages the natural trade-off between memory and compute, storing only those activations that would incur a high recomputation cost lioke the FFN layers.
* **Efficient Backpropagation**: By reducing the amount of recomputation during backpropagation, the method improves the overall efficiency of the model training process, particularly for large-scale models.

1. **Strength and weakness**

**Strengths:**

* Simple techniques to reduce the need for activation recomputation in backward pass. Also, sequence parallelism does not introduce any further communication overhead.
* The techniques reduce activation memory by 5x and cut execution time overhead from recomputation by over 90%.
* Scalability of the models can be further improved with these techniques.

**Weakness:**

* This parallelization strategy is tailored to NVIDIA GPUs. Evaluation methodology could have included other accelerators.
* The paper doesn't extensively discuss whether these techniques affect model convergence or final performance.
* As you increase the model parameters, communication overhead because of sequence parallelism might become apparent.
* Sequence length of 1M or even beyond might still introduce memory challenges

1. **Future Directions**

* Future work could explore dynamic techniques that adjust which activations to store during training based on the current state of the model, rather than making static decisions upfront.
* Evaluation on different accelerator could give valuable insights on effectiveness of the sequence parallelism and selective activation recomputation.
* Future work could explore dynamic techniques that adjust which activations to store during training based on the current state of the model, rather than making static decisions upfront.